

**IN THE SPECIFICATION**

Please replace paragraph [0017] of page 4 with the following amended paragraph:

**[0017]** ~~Figure 2 illustrates~~ Figures 2A and 2B illustrate a mode register of one embodiment of the present invention;

Please replace paragraph [0067] of page 11 with the following amended paragraph:

**[0067]** The Mode Register 148 is used to define the specific mode of operation of the synchronous flash memory. This definition includes the selection of a burst length, a burst type, a CAS latency, and an operating mode, as shown in ~~Figure 2~~ Figures 2A and 2B. The Mode Register is programmed via a LOAD MODE REGISTER command and retains stored information until it is reprogrammed. The contents of the Mode Register may be copied into the NVMode Register 147. The NVMode Register settings automatically load the Mode Register 148 during initialization. Details on ERASE NVMODE REGISTER and WRITE NVMODE REGISTER command sequences are provided below. Those skilled in the art will recognize that an SDRAM requires that a mode register must be externally loaded during each initialization operation. The present invention allows a default mode to be stored in the NV mode register 147. The contents of the NV mode register are then copied into a volatile mode register 148 for access during memory operations.

Please replace paragraph [0102] of page 26 with the following amended paragraph:

**[0102]** The synchronous flash memory array architecture is designed to allow sectors to be erased without disturbing the rest of the array. The array is divided into 16 addressable “blocks” that are independently erasable. Figure 15 illustrates a memory address map of one embodiment of the memory having two boot sectors. By erasing blocks rather than the entire array, the total device endurance is enhanced, as is system flexibility. Only the ERASE and BLOCK PROTECT functions are block oriented. The 16 addressable blocks are equally divided into four banks 104, 106, 108 and 110 of four blocks each. The four banks have simultaneous read-while-

write functionality. An ISM WRITE or ERASE operation to any bank can occur simultaneously to a READ operation to any other bank. The Status Register 134 may be polled to determine which bank is under ISM operation. The synchronous flash memory has a single background operation ISM to control power-up initialization, ERASE, WRITE, and PROTECT operations. Only one ISM operation can occur at any time; however, certain other commands, including READ operations, can be performed while the ISM operation is taking place. An operational command controlled by the ISM is defined as either a bank-level operation or a device-level operation. WRITE and ERASE are bank-level ISM operations. After an ISM bank operation has been initiated, a READ to any location in the bank may output invalid data, whereas a READ to any other bank will read the array. A READ STATUS REGISTER command will output the contents of the Status Register 134. The ISM status bit will indicate when the ISM operation is complete ( $SR7 = 1$ ). When the ISM operation is complete, the bank will automatically enter the array read mode. ERASE NVMODE REGISTER, WRITE NVMODE REGISTER, BLOCK PROTECT, DEVICE PROTECT, and UNPROTECT ALL BLOCKS are device-level ISM operations. Once an ISM device-level operation has been initiated, a READ to any bank will output the contents of the array. A READ STATUS REGISTER command may be issued to determine completion of the ISM operation. When  $SR7 = 1$ , the ISM operation will be complete and a subsequent ISM operation may be initiated. Any block may be protected from unintentional ERASE or WRITE with a hardware circuit that requires the RP# pin be driven to VHH before a WRITE or ERASE is commenced, as explained below.

Please replace paragraph [0132] of page 39 with the following amended paragraph:

**[0132]** Figure 23 is a flow chart of a block unprotect sequence according to one embodiment of the present invention. The sequence includes loading the command register (code 60H), and receiving an active command and a row address. The memory then determines if the memory device is protected. If it is not protected, the memory determines if the boot locations (~~blocks 0 and 15~~)(blocks 0 and 15, shown as elements 210, 220 of Figure 15) are protected. If none of the blocks are protected the memory performs a write operation (D0H) to the block and monitors the status register for completion. An optional status check can be performed and the memory is

placed in an array read mode. If the device is protected, the erase is not allowed unless the RP# signal is at an elevated voltage (VHH). Likewise, if the boot locations are protected, the memory determines if all blocks should be unprotected.

Please replace paragraph [0140] of page 41 with the following amended paragraph:

**[0140]** A data buffer 330 can be coupled to the data communication connections to manage the bi-directional data communication. This buffer can be a traditional FIFO or pipelined input/output buffer circuit. The write latch is coupled between the data buffer and the memory array to latch data provided on the data communication connections. Finally, ~~a control circuit a~~ control circuit 340 is provided to manage the read and write operations performed on the array.